A COMPARATIVE ANALYSIS OF CROSS-CORRELATION MATCHING ALGORITHMS USING A PYRAMIDAL RESOLUTION APPROACH

NUNO ROMA
Instituto Superior Técnico / INESC-ID
Rua Alves Redol, 9 - 1000-029 Lisboa - PORTUGAL
E-mail: Nuno.Roma@inesc-id.pt, *URL:* http://sips.inesc-id.pt/~nfvr

JOSÉ SANTOS-VICTOR
Instituto Superior Técnico / ISR
E-mail: jasv@isr.ist.utl.pt, *URL:* http://www.isr.ist.utl.pt/~jasv

JOSÉ TOMÉ
Instituto Superior Técnico / INESC-ID
E-mail: Jose.Tome@inesc-id.pt

Disparity map estimation is often regarded as one of the most demanding operations in computer vision applications. Several algorithms have been proposed to solve this problem. With such a number of distinct approaches, the question of choosing the most suitable algorithm for a given application is often raised. Few and limited resources can be found in the literature covering this problem. In the following paragraphs it will be presented a comparative analysis of the performance and characteristics of a set of similarity measure algorithms proposed in the literature in the past few years. The obtained results can be regarded as an extremely valuable basis for selecting the most suitable registration algorithm for a given application. The study was focused on the analysis of two distinct aspects: the final matching error and the computational load of each of the considered correlation functions. Besides this comparative study, the advantages of using a pyramidal resolution approach were also considered. This scheme has proved to be effective in reducing the overall computation time and the required number of arithmetic operations, having an insignificant loss in the final matching error.

1 Introduction

During the past few years, the estimation of the disparity field of an image sequence has been playing an increasingly important role in a large number of computer vision applications, such as video coding, multi-view image generation, camera calibration, 3D reconstruction from stereo image pairs, object recognition, visual control and motion estimation[1,2]. The main purpose of this research is to present a quantitative and comparative analysis of several cross-correlation similarity measures used in disparity estimation, applying a pyramidal resolution approach. Matching error and computational load mea-

sures will be used to compare the several registration algorithms, as well as the different possible configurations of the hierarchical processing scheme.

By definition, disparity is the difference between the coordinates of two matched points. The disparity map is composed by a dense field of disparity vectors, one for each matched pixel or group of pixels. However, depending on the considered application, this definition can have some slightly different interpretations. While in video coding this map is a set of motion vectors computed using two distinct images which correspond to two different time instants, in 3D reconstruction it is computed using two images corresponding to two different points of view of the same scene. In this last case, the obtained vectors provide 3D perception by inferring depth information from the scene, thus enabling 3D reconstruction of objects[3]. To make this possible, a high-resolution estimation of disparity vectors of all image pairs is often required, leading to the computation of high accuracy disparity maps.

Several different approaches have been proposed to solve this correspondence search problem[1]. Most of them can be classified in three distinct groups:

- *Feature based algorithms*, where some predefined features are extracted from both images, such as edge segments or contours, to compute the correspondence matching field;
- *Area based algorithms*, where registration algorithms using cross-correlation based similarity measures[4] are used to find the block of image pixels which best matches the one being analyzed in the other image;
- *Optical-flow algorithms*, relying on the relation between photometric correspondence vectors and spatiotemporal derivatives of luminance in an image sequence[5].

Although *feature based algorithms* and *optical-flow algorithms* have been object of an intense research during the last few years, conventional area based algorithms still remain very popular and will assume an important position in the next future in correspondence computation tasks. The main reason for this fact relies on their simple and straightforward implementation, well suited for parallel implementations based on VLSI circuits or Digital Signal Processors (DSP), as well as their robustness against certain image transformations. Furthermore, they directly provide the dense disparity map, whereas in feature based approaches an interpolation step is required if a dense map of the scene is desired. However, they also have some drawbacks that should not be ignored. The most serious one is concerned with the significant computational load, associated to the computation of the dense matching field. To circumvent this disadvantage, a hierarchical processing scheme is now proposed. Furthermore, although several similarity measure functions have been proposed in the past, few and limited resources can be found in

the literature comparing and characterizing them. Since the final performance can be greatly influenced by a correct selection of the registration algorithm, a complete and exhaustive comparison of these functions urges to be performed. With such a study, such as the one presented in this research work, it will be easier to select the most suitable similarity function to be used in a given application, depending on the desired matching error level or on the available computational resources.

This research work is organized as follows. In section 2 area based matching algorithms will be described, as well as their major advantages and disadvantages. section 3 will present the several registration algorithms covered by this research. The pyramidal processing scheme will be described in section 4. In section 5 it will be presented the used evaluation methods and the obtained experimental results. Section 6 concludes the presentation.

2 Area Based Matching Algorithms

In area based matching algorithms, rectangular blocks of pixels from a pair of $M \times N$ images (left and right images) are compared and matched (see figure 1). For each block of the right image (reference window), a corresponding block in the left image is sought, using a given similarity measure as main criteria (see figure 2). During the search process, the left image block is displaced by integer increments (c, l) around a predefined region (search window), and an array of similarity scores $d(c, l)$ is computed (see eq. 1). The position (c_M, l_M) of the moving block corresponding to the maximum computed value of the considered similarity function $\odot$ for that search window is selected and chosen to obtain the optimum disparity vector corresponding to that reference window. Hence, a matching dense field $D(x, y)$ is computed by using as many overlapping search windows as the number of pixels of the image, thus obtaining a disparity vector for each pixel (see eq. 3).

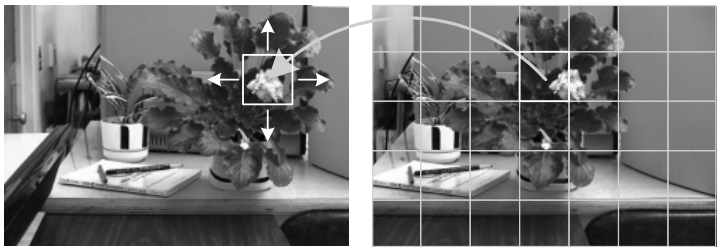


Figure 1. Disparity map estimation process.

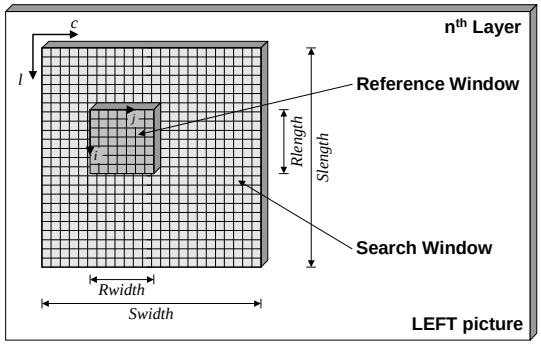


Figure 2. Searching procedure.

$$d\,(c,l) = \sum_{v=0}^{R_{width}} \sum_{u=0}^{R_{length}} R\,(u,v) \odot S\,(c+u, l+v) \tag{1}$$

$$d\,(c_M, l_M) = \arg\max_{(c,l)} \{\, d\,(c,l) \;:\; 0 \le c < S_{width}\,;\; 0 \le l < S_{length}\,\} \tag{2}$$

$$\mathbf{D}\,(x,y) = \{\, d_{xy}\,(c_M, l_M) \;:\; 0 \le x < M\,;\; 0 \le y < N\,\} \tag{3}$$

Several similarity functions have been proposed in the literature[4]. The selection of a particular function is usually based on its computational load, algorithmic simplicity and achieved performance. Similarity measures based on cross-correlation, normalized cross-correlation, sum of squared differences and sum of absolute differences are often chosen. However, other measures based on co-occurrence matrices have recently been proposed[8].

Moreover, the selection of the reference and search windows width is not a simple and trivial task[7,1]. In fact, while the probability of a mismatch usually decreases when the reference window is enlarged, using large windows often leads to an accuracy loss, since the influence of image differences is greatly aggravated with the increase of the considered area, as will be shown in section 5. On the other hand, the size of the search window influences the maximum allowed value of the resulting disparity vector. The greater the search window, the greater the allowed mobility of a given pixel. Therefore, this window should be wide enough so that it comprises the correspondent block of a given reference window. Furthermore, a difficult and important trade-off must be done when selecting these window widths, since the computational load and processing time usually increase linearly with their areas.

Therefore, a compromise must often be done, by adjusting these parameters according to the image size and its contents.

Consequently, disparity estimation of a dense matching field is usually considered to be one of the most challenging tasks of 3D reconstruction. This fact can be explained by several reasons, such as:

- The dimension of the solution space is extremely large, since each image pixel can be matched with any pixel of the other image.
- It is involves computational demanding algorithms.
- Its accuracy is extremely dependent on the photometric characteristics of the images being analyzed such as texture, luminance and contrast, and the noise conditions associated to the acquisition camera system. This task can be particularly influenced in images with lack of information, e.g., regions with constant bright, horizontal edges and repetitive patterns.
- The existence of partially or totally occluded objects in the image pair can give rise to disparity errors with difficult solution.

The estimation of dense disparity maps is usually performed by taking in consideration a set of constrains relating the two images being analyzed. One of these constrains is the so-called *Constant Image Brightness* (CIB) assumption, which states that a matched pixel pair should have an equal luminance value[1]. Therefore, the main problem consists in finding the pixel whose neighborhood best matches the region being analyzed (see figure 3). Theoretically, all points of the other image can be accepted as possible candidates to this matching process. However, there is at most only one pixel which corresponds geometrically. Consequently, the CIB constraint alone is regarded as being insufficient for the estimation of dense disparity maps, being often considered an ill-posed problem. This task can be simplified by using some more constrains, making it possible to reduce the dimension and ambiguity of this problem, thus decreasing the total processing time. Some of the constrains more frequently used are[2]:

- *Epipolar Constraint*: provides a conversion from a 2D search into a 1D

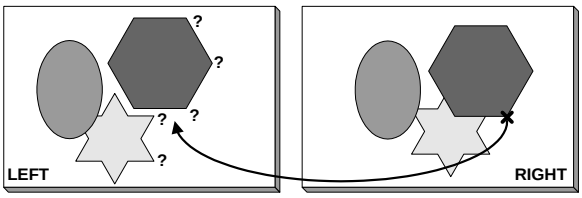


Figure 3. Any point of the left image is a possible candidate to the matching process.

search, by imposing that the matched points must lie on the corresponding epipolar line of the two images;

- *Unicity*: imposes that each pixel can have, at most, one correspondent pixel in the other image;
- *Smoothness*: imposes a continuous and smooth variation of the disparity field;
- *Order Constraint*: forces the order of the points belonging to an epipolar line to remain the same;
- *Disparity Gradient*: limits the allowed variation of the disparity values.

3 Cross-Correlation Algorithms

The set of similarity functions considered in the comparative analysis are presented in table 1, as well as their definition expressions[4]. Due to their potentially robust real-time operation and their moderate requirements in terms of hardware and software resources, only correlation type algorithms were considered in this study. In this table, $R(u,v)$ denotes a reference window pixel, while $S(c,l)$ denotes a search window pixel, $\overline{R}$ the local mean of the reference window ($\overline{R} = \sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} R(u,v)$), and $\overline{S(c,l)}$ the pixel mean in the block of the search window being compared ($\overline{S(c,l)} = \sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} S(c+u,l+v)$).

Table 1: Cross-Correlation Algorithms.

Correlation Name	Definition
Simple Cross-Correlation **SCC(c, l)**	$\sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} R(u,v) \cdot S(c+u,l+v)$
Normalized Cross-Correlation **NCC(c, l)**	$\frac{\sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} R(u,v) \cdot S(c+u,l+v)}{\sqrt{\sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} R^2(u,v) \cdot \sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} S^2(c+u,l+v)}}$
Zero Mean Normalized Cross-Correlation **ZNCC(c, l)**	$\frac{\sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} \left(R(u,v)-\overline{R}\right) \cdot \left(S(c+u,l+v)-\overline{S(c,l)}\right)}{\sqrt{\sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} \left(R(u,v)-\overline{R}\right)^2 \cdot \sum_{v=0}^{R_{length}} \sum_{u=0}^{R_{width}} \left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}}$

Table 1: (continued)

Correlation Name	Definition
Moravec **MOR**(**c**, **l**)	$\frac{2\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-\overline{R}\right)\cdot\left(S(c+u,l+v)-\overline{S(c,l)}\right)}{\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-\overline{R}\right)^2+\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}$
Normalized Zero Mean Sum of Squared Differences **NZSSD**(**c**, **l**)	$\frac{\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left[\left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right]^2}{\sqrt{\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-\overline{R}\right)^2\cdot\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(S(c+u,l+v)-\overline{S(c,l)}\right)^2}}$
Sum of Squared Differences **SSD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-S(c+u,l+v)\right)^2$
Sum of Absolute Differences **SAD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left\|R(u,v)-S(c+u,l+v)\right\|$
Normalized Sum of Squared Differences **NSSD**(**c**, **l**)	$\frac{\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-S(c+u,l+v)\right)^2}{\sqrt{\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}R^2(u,v)\cdot\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}S^2(c+u,l+v)}}$
Zero Mean Sum of Squared Differences **ZSSD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left[\left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right]^2$
Zero Mean Sum of Absolute Differences **ZSAD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left\|\left(R(u,v)-\overline{R}\right)-\left(S(c+u,l+v)-\overline{S(c,l)}\right)\right\|$
Locally Scaled Sum of Squared Differences **LSSD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left(R(u,v)-\frac{\overline{R}}{\overline{S(c,l)}}\cdot S(c+u,l+v)\right)^2$
Locally Scaled Sum of Absolute Differences **LSAD**(**c**, **l**)	$\sum_{v=0}^{R_{length}}\sum_{u=0}^{R_{width}}\left\|R(u,v)-\frac{\overline{R}}{\overline{S(c,l)}}\cdot S(c+u,l+v)\right\|$

The functions SCC, NCC, $ZNCC$ and MOR are pure similarity measures, since the best match corresponds to the maximum value obtained with these functions. In contrast, the functions $NZSSD$, SSD, SAD, $NSSD$, $ZSSD$, $ZSAD$, $LSSD$ and $LSAD$ represent difference or dissimilarity func-

tions and the best match is obtained when the value returned by these functions is the minimum. Some of these functions are normalized versions with respect to the mean and standard deviation of the SCC, SSD and SAD functions. The objective is to make these registration algorithms insensitive to changes of the brightness and contrast of $R(u,v)$ and $S(c,l)$ values[9]. Furthermore, in order to overcome possible distortions of these measures in the vicinity of the image bounds, a block normalization is often performed, by dividing each correlation result by the area of the correspondent reference window:

$$\hat{d}(c,l) = \frac{1}{R_{width} \times R_{length}} \cdot d(c,l) \tag{4}$$

Although the described functions present evident analogies, the correspondent computational load and hardware requirements can be significantly different. While with the SCC, the simplest function, it is only necessary to perform $R_{length} \times R_{width}$ multiply-and-accumulate (MAC) operations, arithmetic units capable of performing squared-root operations ($ZNCC$, $NZSSD$, $NSSD$), absolute-value operations (SAD, $ZSAD$, $LSAD$) or integer divisions (NCC, $ZNCC$, MOR, $NZSSD$, $NSSD$, $LSSD$, $LSAD$) are requited in other functions. These requirements are often an important aspect when selecting the most suitable similarity measure function for a given implementation, as will be further illustrated in section 5.

4 Pyramidal Processing Scheme

In order to obtain a high-accuracy disparity computation, it is important to use reference and search windows large enough to provide the computation with the correct match even when pixel pairs present significant disparity values. However, the computational effort of this correspondence search increases significantly when the area of these windows increase. Furthermore, larger windows are usually associated to longer computation times.

One form of solving these implementation issues is to use a hierarchical approach, by using a pyramidal processing scheme like the one depicted in figure 4[1,6]. With this technique, the matching process is done in a multi-layered fashion and is based on a coarse-to-fine approach, providing significant functional and computational advantages[10]. The left and right images are successively down-sampled by a factor of 2, using a decimation function to obtain lower resolution versions of the original image pair. The original images represent level 0 and images resolution is decreased with the pyramid level. Therefore, the pyramid may be viewed as a 4D data structure, where the

intensity of pixels is a function $f(l, x, y)$ with 3 arguments: a level designator (l) and a pair of coordinates (x, y).

The matching estimation process is started at level L. This ensures that the earlier correlations are performed with the gross image features rather than with the details. The matching of these gross features will be used to guide the later high-resolution searches, thus achieving more accurate matches of these features and of the surrounding details. After this set of low-resolution pictures has been processed, the obtained disparity map is interpolated to the resolution of level $L-1$. These disparity values are then used as an initial estimate for the computation of the disparity map of this level (see figure 4). This process continues until estimation of the disparity map corresponding to full resolution (at level 0), is performed. Therefore, to estimate the disparity field using several resolution layers, it is only necessary to repetitively

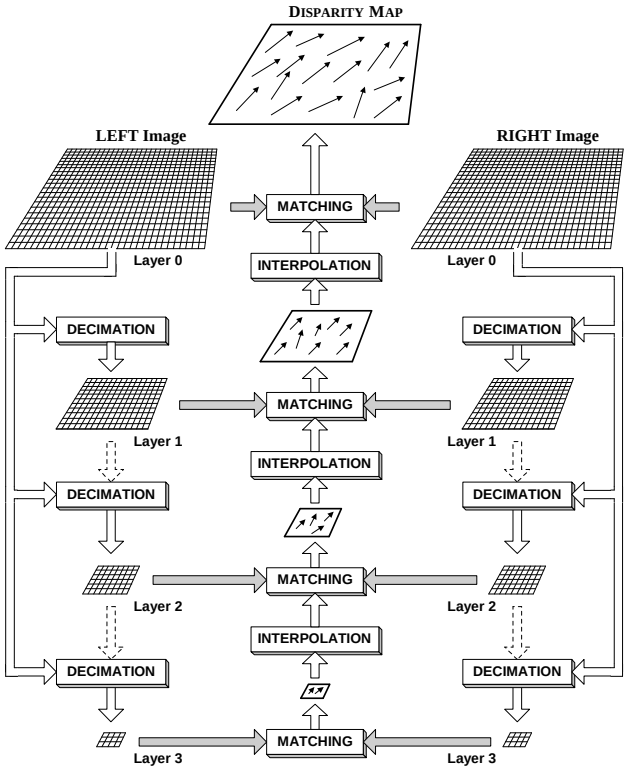


Figure 4. Pyramidal Processing Scheme.

apply the same algorithm to each of the considered levels. Moreover, by using this scheme it is possible to use the same small search and reference windows along all the layered processing scheme. Consequently, each time the images resolution is increased, the coverage of these windows is reduced by a factor of 4, thus providing a gradual refinement of the matching process and a greater treatment of the details. This makes it possible to obtain accurate disparity values and significant coverage areas, which could only be obtained with the usage of larger and more time consuming windows in a single layered processing architecture.

Some authors have proposed an additional strategy to speed up the matching estimation, by working with sub-images rather than processing the entire image[11]. However, although with this solution the required memory space is lower, it involves an additional overhead in the whole processing scheme.

In the next subsections it will be described, in a more detailed way, several important aspects of the pyramidal processing scheme.

4.1 Number of Layers

One of the most critical decisions that usually arise when using a pyramidal processing scheme is concerned with the number of layers used in the structure. By increasing the number of layers it is possible to use smaller reference and search windows, thus leading to faster estimations of the dense disparity field. On the other hand, important features and other types of image information required to the matching process can be lost or distorted when too coarse resolutions are used, giving rise to critical problems in the search process[12]. Moreover, multi-scale image representations should be consistent, since features at different resolutions may be correlated[6,12]. Therefore, significant features at different layers should not randomly appear or disappear when resolution is increased.

With careful design of the decimation and interpolation blocks of the hierarchical scheme, satisfactory results can be obtained. However, image size and its contents should always be considered in the decision of the number of layers used by the hierarchical structure.

4.2 Decimation Function

A pyramidal structure is usually implemented by sub-sampling the original image. In order to fulfill the Nyquist theorem, a low-pass filtering of the original image is required to be first performed. The filter implemented in the developed system is a gaussian filter centered at $\overline{m} = (m_x, m_y)$ and with

variance σ^2, having the impulse and frequency responses given by eq. 5, and a 3dB bandwidth given by eq. 6.

$$h(\overline{x}) = \frac{1}{\sqrt{2\pi}.\sigma} \cdot e^{-\frac{(\overline{x}-\overline{m})^2}{2.\sigma^2}} \qquad \longrightarrow \qquad H(\overline{f}) = e^{-2\pi^2\sigma^2\overline{f}^2} \tag{5}$$

$$BW_{-3dB} = \left.\frac{-\ln(\alpha)}{2\pi^2\sigma^2}\right|_{\alpha=\frac{1}{\sqrt{2}}} = \frac{\ln(2)}{4\pi^2\sigma^2} \tag{6}$$

Therefore, the set of $(M_L \times N_L)$ pixels that compose the image layer of level L $(f_L(i,j))$ can be obtained by performing a 2D convolution of the image matrix of level $L-1$ with $(M_{L-1} \times N_{L-1})$ pixels $(f_{L-1}(i,j))$, with the impulse response of the gaussian filter $h(x,y)$ followed by the application of the sub-sampling process.

The efficiency of this algorithm can be greatly improved by noting that $(M_{L-1} \times N_{L-1})$ filter results are computed in this stage and only $(M_L \times N_L) = (M_{L-1} \times N_{L-1})/4$ pixels are used after the sub-sampling process. If this 25% efficiency is considered in a K layered pyramid, it is possible to conclude that only $(0.25)^{K-1}$ % of the computations are actually used. To circumvent this limitation a different approach has been used: only the pixels which are used in the next sampling process to obtain the new desired layer are actually computed. This procedure can be regarded as a combination of the filtering and sub-sampling phases, thus avoiding the computation of unnecessary results.

To avoid a gradual decrease of the precision of the filter results as a consequence of multiple and cascaded quantization steps, the original image has been always used in all filter operations and at all implemented layers. This implies a gradual decrease of the filter bandwidth when the pyramidal algorithm descends to lower layers. As it was shown in eq. 6, this can be easily done by increasing the filter variance σ^2 with the level number. In the

Figure 5. Application of a 4 layered decimation function to *lenna* test image.

developed system, it was considered a variance given by $\sigma_L^2 = 2 \times \sigma_{L-1}^2$ in the computation of the filtered layers, using a window of size $6 \times \sigma_L^2$. This window dimension guarantees that more than 99% of the area of the 2D gaussian filter of eq. 5 lies within this window.

In figure 5 it is presented the result obtained with the application of the described decimation function to the 256×256 *lenna* test image.

4.3 Matching Process

The matching process was performed using the set of correlation based similarity functions described in section 3 and presented in table 1. To achieve an efficient implementation of each of these algorithms, some manipulations of the expressions presented in table 1 have been performed in order to obtain the final result in the minimum number of steps, using the minimum number of operations[11]. This is illustrated in eq. 7 through 10 for the general case of a zero-mean normalized cross-correlation using the following simplified nomenclature described in section 3: $R = R(u,v)$, $S = S(c+u, l+v)$, $\overline{S} = \overline{S(c,l)}$, $L = R_{length}$, $W = R_{width}$, $\sum\sum \alpha = \sum_{v=0}^{L} \sum_{u=0}^{W} \alpha$.

$$d(c,l) = \frac{cov_{c,l}(f,g)}{var_{c,l}(f) \times var_{c,l}(g)} \tag{7}$$

$$\begin{aligned} var(f) &= \sum\sum \left(R - \overline{R}\right)^2 = \sum\sum \left(R^2 - 2R\overline{R} + \overline{R}^2\right) \\ &= \sum\sum R^2 - 2 \cdot \frac{\left(\sum\sum R\right)^2}{L.W} + \left(\frac{\sum\sum R}{L.W}\right)^2 \times L.W \\ &= \sum\sum R^2 - \frac{\left(\sum\sum R\right)^2}{L.W} \end{aligned} \tag{8}$$

$$var(g) = \sum\sum S^2 - \frac{\left(\sum\sum S\right)^2}{L.W} \tag{9}$$

$$\begin{aligned} cov_{c,l}(f,g) &= \sum\sum \left(R - \overline{R}\right)\left(S - \overline{S}\right) \\ &= \sum\sum R.S - \overline{R}\sum\sum S - \overline{S}\sum\sum R + L.W \times \overline{R}.\overline{S} \\ &= \sum\sum R.S - \frac{\sum\sum R}{L.W} \cdot \sum\sum S - \frac{\sum\sum S}{L.W} \cdot \sum\sum R + \\ &\quad L.W \times \frac{\sum\sum R}{L.W} \cdot \frac{\sum\sum S}{L.W} \\ &= \sum\sum R.S - \frac{\sum\sum R \cdot \sum\sum S}{L.W} \end{aligned} \tag{10}$$

Consequently, for the majority of these algorithms, it is only necessary to compute the values of $\sum_{v=0}^{L}\sum_{u=0}^{W} R$, $\sum_{v=0}^{L}\sum_{u=0}^{W} S$, $\sum_{v=0}^{L}\sum_{u=0}^{W} R.S$, $\sum_{v=0}^{L}\sum_{u=0}^{W} R^2$ and $\sum_{v=0}^{L}\sum_{u=0}^{W} S^2$ in one single step. Besides these manipulations, the block normalization of eq. 4 was also performed.

4.4 Interpolation

As it was described in section 4, the several disparity maps estimated in the lower levels of the pyramidal structure are used as an initial estimate of the disparity fields of subsequent higher levels, following a classic coarse-to-fine approach. However, before these initial estimates can be used, a scaling up operation is required to be performed on the disparity map obtained from the previous layer to conform its dimension and its vectors with the new layer resolution. This function was implemented using a bilinear interpolation algorithm based on the computation of the mean disparity value of the group composed by 4 or 2 neighbor disparity vectors corresponding to the set of pixels belonging to a 3×3 interpolation window.

4.5 Disparity Maps

The final result of this hierarchical processing scheme is a dense disparity map. This map can be seen as a $(M_0 \times N_0)$ array, where each element is a data structure composed by 3 values:

- Disparity value along the xx axis.
- Disparity value along the yy axis.
- Similarity measure value of the correspondent pair of pixels.

Since there is a direct relation between the obtained correlation values and the achieved matching performance, the similarity measure sub-array can be regarded as a confidence map of the final result. Therefore, it can be used to select the pixel coordinates corresponding to the best match of the whole process.

5 Experimental Results

The comparative analysis presented in this research was based on a software implementation of the described algorithms using the object oriented language C++, running on Linux and Windows NT workstations. In the following subsections, it will be described the experiment layout and presented the achieved experimental results.

5.1 Experiment Layout

In the performed comparative analysis, two image pairs representing a scene taken at planet Mars and an aerial view of the Pentagon have been used (see figures 6 and 7). These scenes are considered good examples of high textured pictures, well suited for area-based matching algorithms evaluation. In the several computations carried out along this research, two main aspects were used to assess the several algorithms: *matching error* and *computational load*.

To obtain a fair comparison of the obtained disparity maps, the final registration error of each similarity function has been used to assess the result of the several algorithms. This common measure was estimated by computing, for each pixel of the right image, the sum of the squared differences (e_{xy}) between all pixels belonging to a rectangular window of size ($K \times L$) of the right image ($f(x, y)$) and all pixels belonging to the corresponding window of

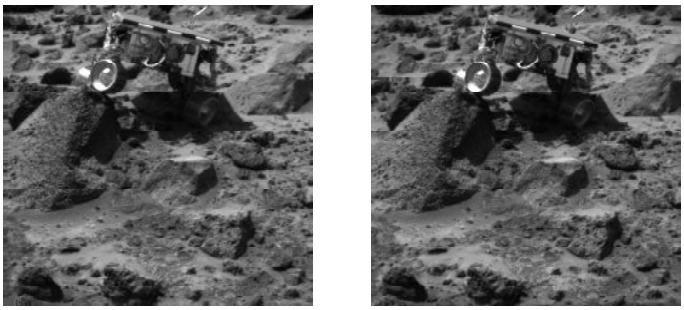


Figure 6. Left and right images of Sojourner, taken from Pathfinder lander camera at planet Mars.

Figure 7. Left and right images of an aerial view of the Pentagon.

the left image ($g(x,y)$), defined with the disparity vector (d_x, d_y):

$$e_{xy} = \sum_{i=-\frac{K}{2}}^{+\frac{K}{2}-1} \sum_{j=-\frac{L}{2}}^{+\frac{L}{2}-1} \left[f(x+i, y+j) - g(x+i+d_x, y+j+d_y) \right]^2 \tag{11}$$

By evaluating the square root of this sum and by dividing it by the area of the considered window, it was obtained a value which quantifies the resultant matching error in the pixel domain (see eq. 12). The matrix $\mathbf{E}(x,y)$ composed by all these $E(x,y)$ values is denominated by error map (see eq. 13). Moreover, by accumulating all these $E(x,y)$ values and normalizing the result with the total image area, it was obtained the value E which best characterizes the performance of a given algorithm (see eq. 14). The E values of the several algorithms were used in the comparative charts presented in the following subsections.

$$E(x,y) = \frac{\sqrt{e_{xy}}}{K \times L} \tag{12}$$

$$\mathbf{E}(x,y) = \{ E(x,y) \ ; \ 0 \leq x < M \ ; \ 0 \leq y < N \} \tag{13}$$

$$E = \frac{\sum_{x=0}^{M} \sum_{y=0}^{N} E(x,y)}{M \times N} \tag{14}$$

In what concerns the computational load, the implemented program was developed in order to provide a statistical study of all arithmetic operations performed along the disparity map estimation, namely, *multiplications*, *additions* and *other* similarity measure specific functions, such as square roots, absolute values and integer divisions.

Furthermore, the performed analysis was also focused on the study of different aspects related to matching estimation using a pyramidal structure. The complete set of similarity functions presented in table 1 was used in this analysis, making it possible to evaluate the specific characteristics of each function in terms of final disparity error and computational load. The parameters considered were:

- *Pyramid depth*, i.e., the number of layers used in the hierarchical scheme;
- *Pixel mobility*, defining the maximum allowed value of each disparity vector, and controlled by adjusting the *search window width* parameter;
- *Reference window width*;
- *Computational load distribution* among the several stages of the system.

In the following subsections, it is presented the set of comparative results achieved using the Mars stereo image pair. These results are entirely similar to those obtained using other test images in what concerns all the considered aspects described above.

5.2 Disparity Maps

In figure 8 it is presented a graphical representation of the components of the disparity vectors along the xx and yy axis. These matrices were obtained applying the $ZNCC$ similarity function with a 2 layered pyramidal structure, a 64 pixels width search window and a 13 pixels width reference window to the Mars stereo image pair. The observed gradual increase of the disparity vectors amplitude in the y direction along the xx axis conforms with the expected behavior. Since these vectors correspond to image points farther from the camera system position, they present more significant coordinates differences. The representation shown in figure 9 presents these matrices in a more intuitive way.

In figure 10 it is presented the correlation map corresponding to this configuration and the error map obtained from eq. 14. In this figure it is possible to distinguish image areas with significant higher values of the disparity error, namely, at the borders of the graph. This higher values are usually due to non-overlapping regions of the image pair. Some of the peaks found in the middle of the error map are often caused by matching mistakes or occluded objects. Several solutions have been proposed in the literature to eliminate these matching mistakes[11]. The approach considered in the present research was based on the previous described *Smoothness* constraint of the disparity

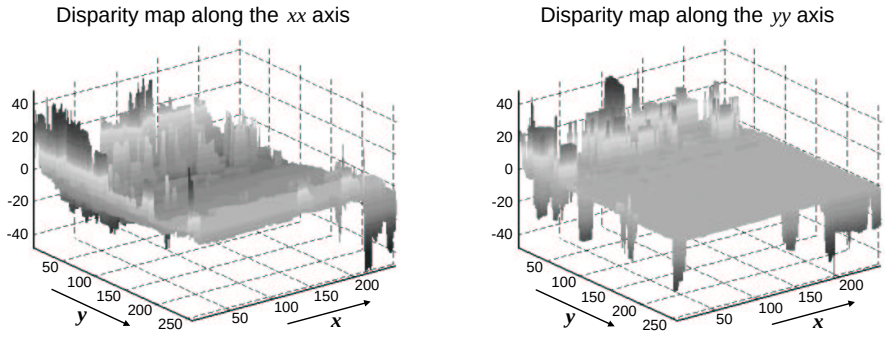


Figure 8. Disparity values along the xx and yy axis.

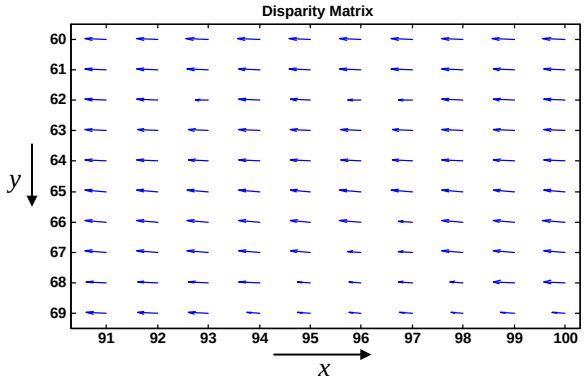


Figure 9. Disparity Matrix.

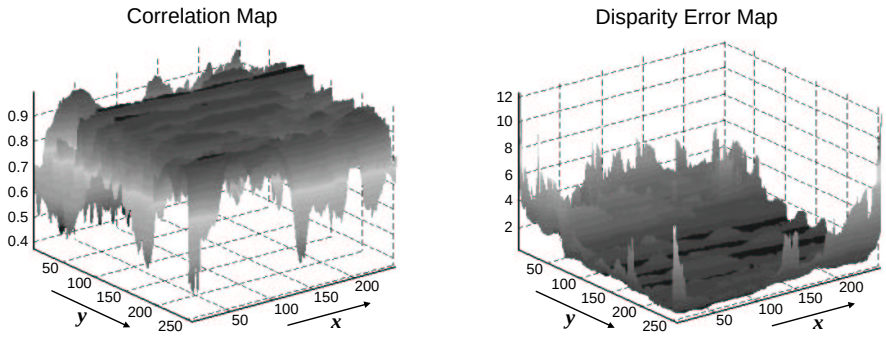


Figure 10. Correlation and Error Maps.

field. It consists in performing a median filtering stage before applying the interpolation step on the disparity field obtained at each level of the pyramidal scheme. The purpose of the filter is to remove abrupt maximum and minimum peaks from the field. Unfortunately, the achieved results have shown effective improvements of only 5% on the final disparity error. Therefore, it was decided to disable this intermediate block of the processing scheme in the subsequent study of the required computational load presented in the next subsections.

5.3 Disparity Error

In figure 11 it is shown the variation of the disparity error with the allowed pixel mobility. In this and in the following charts it has been defined the measure *Full-Mobility (FM)* to designate the largest allowed value of any disparity vector of the estimated map. This value corresponds to the limit situation that occurs when there is a match between the lower-left pixel of one image and the upper-right pixel of the other image. In a non-hierarchical scheme this limit situation would give rise to the usage of a search area four times greater than the original one.

For the majority of the considered configurations it is always possible to find the one corresponding to the best match. In the example given above, it corresponds to the situation where the maximum allowed disparity value is $FM/8$, which corresponds to a 32 pixel width search window. The increase of the error with smaller windows was already expected, since the probability to find the correct matching pixel in the other image decreases. The observed increase of the disparity error when larger search windows were used is probably due to an increase of the image features ambiguity.

Figure 12 represents the variation of the disparity error with the number of levels used pyramidal structure. The presented values evidence better results obtained with fewer hierarchical levels, as a consequence of distortions and losses of essential features required to perform the matching of image regions when too coarse resolutions are used.

Figure 13 describes the influence of the reference window width in the disparity error. The obtained results evidence the existence of an optimal value for this parameter corresponding to the best configuration. For the example given above, reference windows with 11 or 13 pixels width have shown to be more favorable. For smaller windows, the increase of the error value can

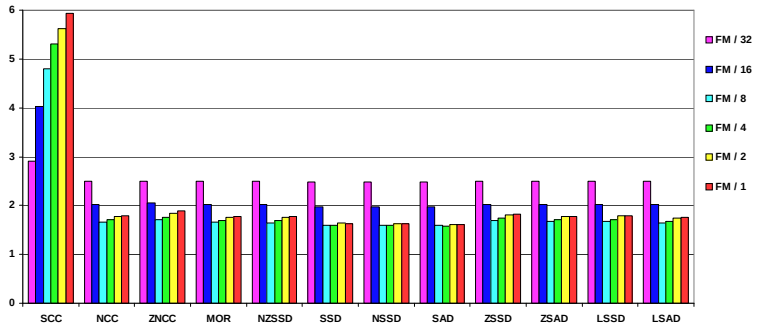


Figure 11. Influence of the allowed pixel mobility in the final disparity error.

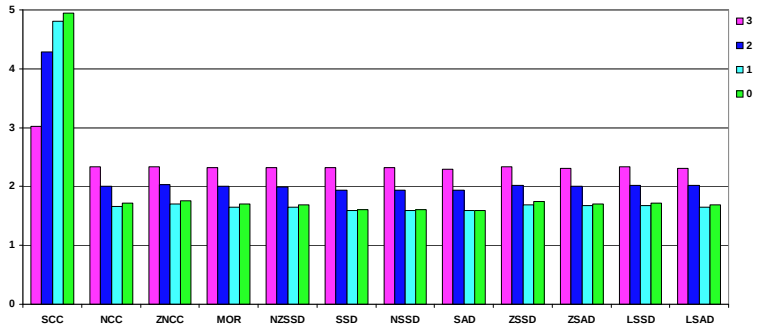


Figure 12. Influence of the pyramid depth in the final disparity error.

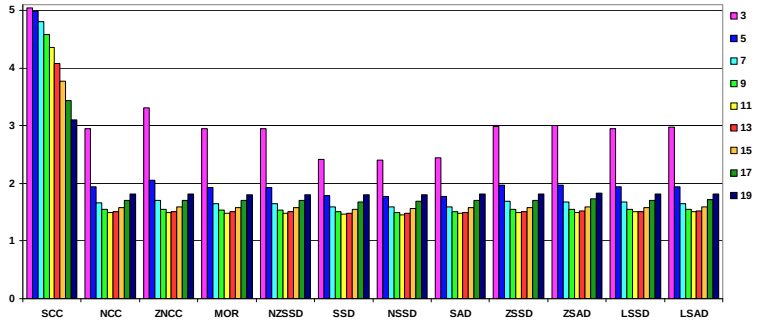


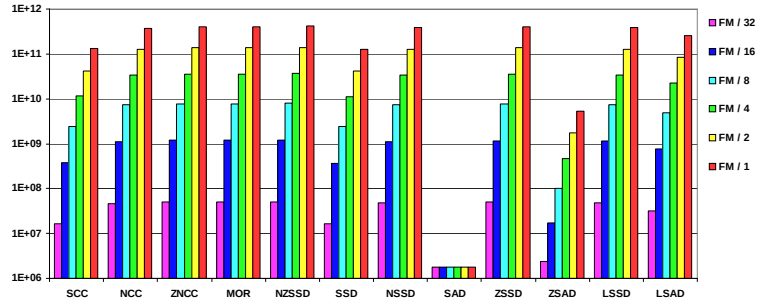
Figure 13. Influence of the reference window width in the final disparity error.

be justified due to a substantial increase of the matching ambiguity, since it becomes very difficult to distinguish the several image features when too small reference windows (such as 3×3 pixels width windows) are used. The increase of the disparity error when larger windows are used can be explained by the increasing influence of the differences found in the image pair due to the different points of view of the acquisition camera system, making it difficult to find the correct match.

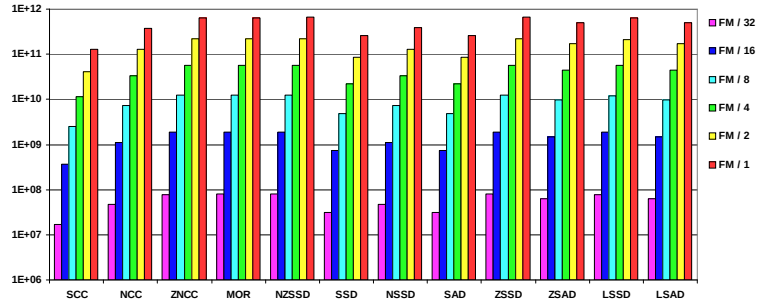
The previous charts also provide useful information to compare the considered similarity functions. From these charts, it is possible to conclude that by using a convenient value for the search window width and reference window width, the final results can be very similar. However, better results can be obtained with SSD, $NSSD$ and SAD similarity functions. The simple cross-correlation function (SCC) provided the worst performance in terms of disparity error. Consequently, its disparity error was not considered in some of the comparisons presented above.

5.4 *Computational Load*

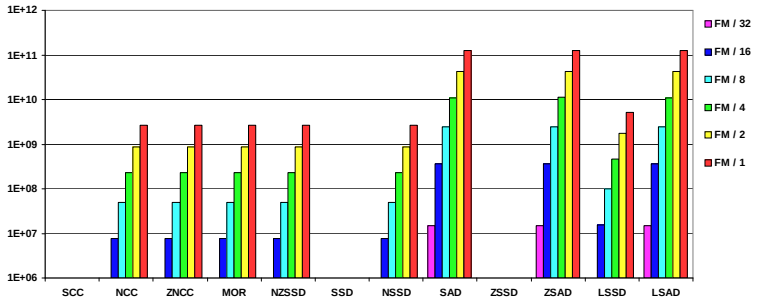
In figure 14 it is presented a statistical comparison of the required number of *multiplications*, *additions* and *other* operations, for different configurations of the maximum allowed pixel mobility parameter. In these and in the following charts, the general designation *Other* corresponds to the computation of:



(a) Multiplications



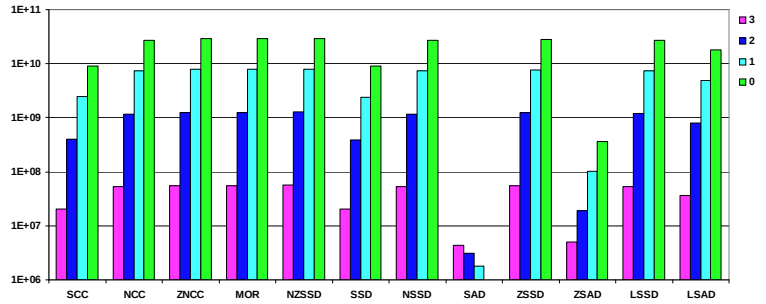
(b) Additions



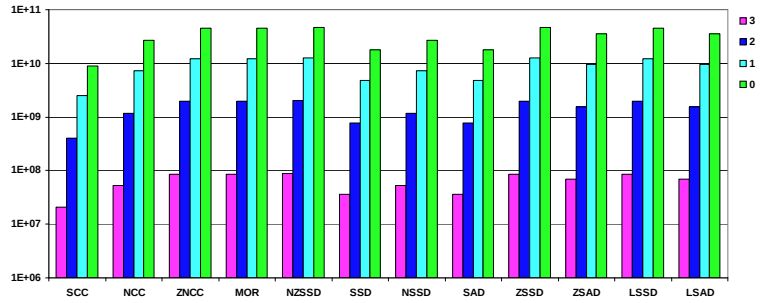
(c) Other

Figure 14. Influence of the allowed pixel mobility in the number of arithmetic operations.

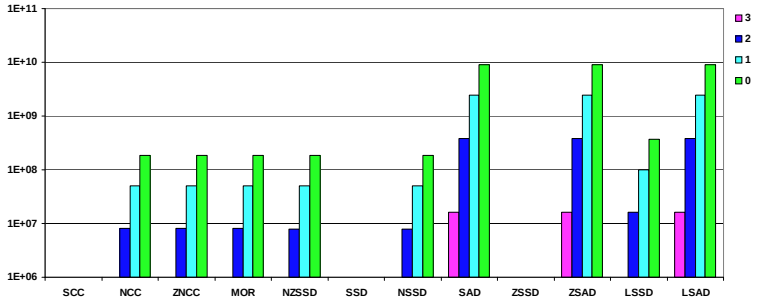
- $Other\,(a,b) = \frac{a}{\sqrt{b}}$ in NCC, $ZNCC$, $NZSSD$ and $NSSD$ similarity functions;
- $Other\,(a,b) = \frac{a}{b}$ in MOR and $LSSD$ similarity functions;
- $Other\,(a) = |a|$ in SAD, $ZSAD$ and $LSAD$ similarity functions.



(a) Multiplications



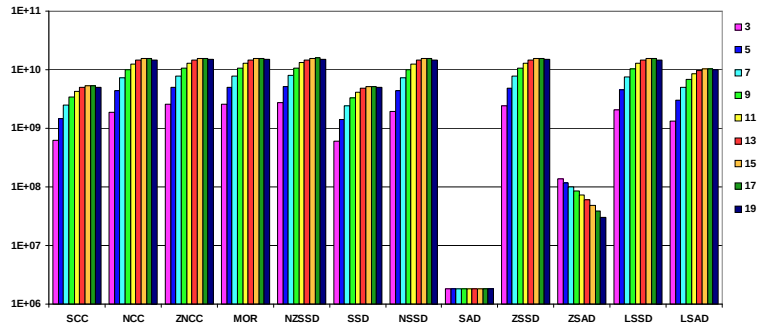
(b) Additions



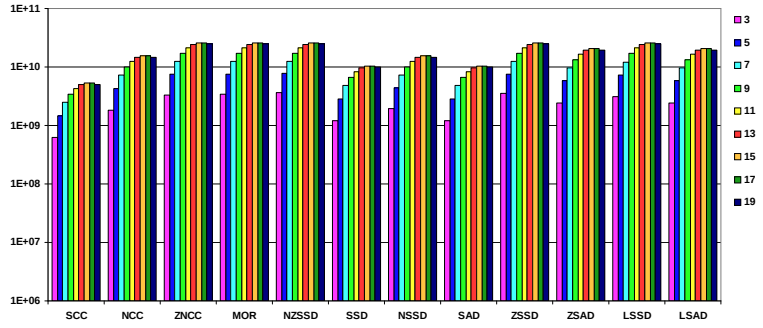
(c) Other

Figure 15. Influence of the pyramid depth in the number of arithmetic operations.

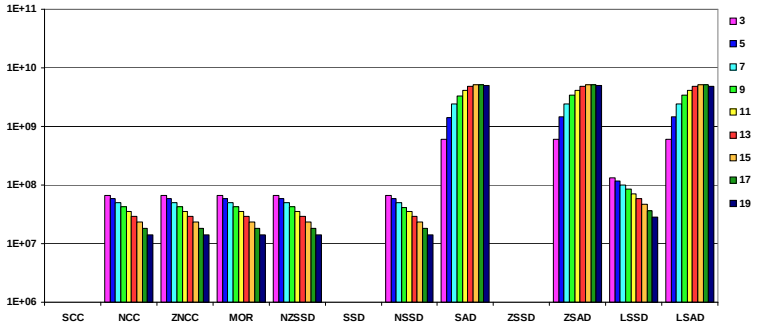
These charts clearly evidence the significant increase of the number of arithmetic operations with the mobility factor (reflected in a correspondent increase of the search window area) is responsible for a performed along the



(a) Multiplications



(b) Additions



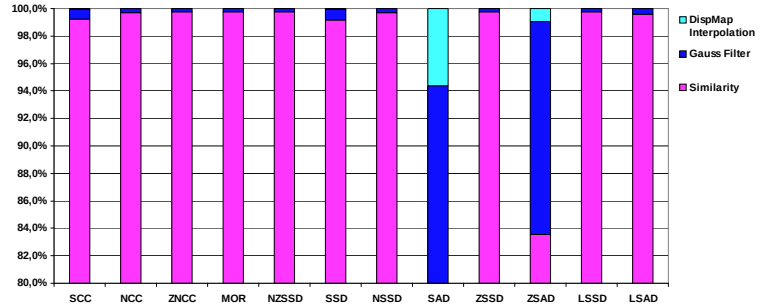
(c) Other

Figure 16. Variation of the number of operations with the width of the reference window.

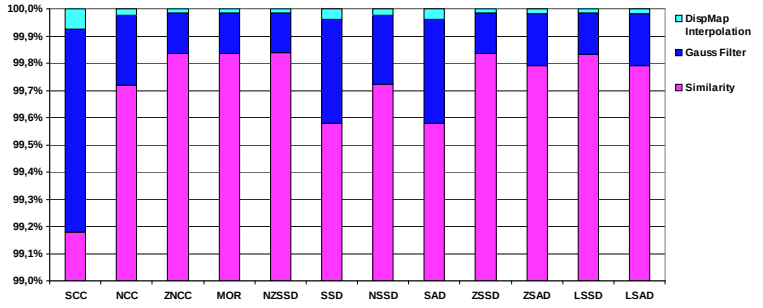
estimation process. As an example, it is possible to verify that a search window corresponding to $FM/1$ implies a computational load 8000 times higher than a window corresponding to $FM/32$.

In figure 15 it is shown a statistical comparison of the required number of arithmetic operations to estimate the dense disparity field using 0 (original resolution), 1, 2 and 3 hierarchical levels. These results evidence the significant advantages of using a hierarchical approach. They allow us to conclude that the computational load of a 3 layered scheme can be 500 times lower than the required by a plain structure approach. The exceptional lower number of multiplications required by SAD similarity function was already expected because this operation is only performed in the filter and interpolation stages of this registration algorithm.

In figure 16 it is presented the variation of the computational load with the width of the reference window, emphasizing the general increase of the



(a) Multiplications



(b) Additions

Figure 17. Distribution of the *multiply* and *add* operations in the several stages of the system.

number of arithmetic operations performed in the matching estimation stage when the reference window is enlarged.

Finally, in figure 17 it is presented the distribution of the global computational load into the three main stages of the system: *Similarity computation*, *Gaussian filtering* and *Disparity map interpolation*. These charts clearly show that the similarity measure computation is responsible for the major part of the operations performed in the estimation process. Moreover, the overhead of the several auxiliary stages required to perform a pyramidal processing scheme has an almost insignificant weight in the overall computational load.

Therefore, the results presented in this subsection can be regarded as a very important comparison basis to select the most convenient similarity function and the most suitable set of parameters of the pyramidal processing scheme to estimate the dense disparity map of a given stereo image pair. Likewise, they can also be regarded as a primordial basis for taking a decision in the critical tradeoff between computational load saving and final disparity error minimization.

6 Conclusion

In this research work it has been presented a comparative analysis of several area based similarity measure functions, using a pyramidal resolution scheme. The similarity functions analyzed constitute a set of twelve different cross-correlation based matching algorithms. Among this set, it has been shown that better results can be obtained when zero-mean normalized similarity functions are used, such as $ZNCC$ and MOR. Dissimilarity functions like SSD and SAD have also proved to give rise to good results.

The presented research has also shown the influence of several parameters of this hierarchical scheme on the overall *Disparity Error* and *Computational Load* performances. Among this set of parameters, the number of layers, the search window width and the reference window width have proved to be significant parameters to achieve a convenient tradeoff between the two referred factors. Moreover, some registration algorithms based in the computation of dissimilarity measures using the absolute value operator (SAD, $ZSAD$ and $LSAD$) have shown to be very cost effective in what concerns the required computational resources. This fact is tied in with the low required number of multiplication operations, usually associated to higher demanding resources, in what concerns arithmetic units and processing time. As a consequence, recent implementations based on VLSI circuits or Digital Signal Processors have adopted this set of registration algorithms.

References

1. A. Redert, E. Hendriks, J. Biemond, "Correspondence Estimation in Image Pairs", *IEEE Signal Processing Magazine*, May 1999, pp. 29-45.
2. A. Arsénio, J. S. Marques, "Performance Analysis and Characterization of Matching Algorithms", *Proc. of the* 5^{th} *International Symposium on Intelligent Robotic Systems*, Stockholm, Sweden, July 1997.
3. M. G. Strintzis, S. Malassiotis, "Object-Based Coding of Stereoscopic and 3D Image Sequences", *IEEE Signal Processing Magazine*, May 1999, pp. 14-28.
4. P. Aschwanden, W. Guggenbühl, "Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms", *in Forstner and Ruweidel, eds.*, *Robust Computer Vision*, pp. 268-282, Wichmann, (1992).
5. E. Grossmann, J. Santos-Victor, "Performance Evaluation of Optical Flow Estimators: Assessment of a New Affine Flow Method, Etienne Grossmann", *VisLab-TR 07/97 - Robotics and Autonomous Systems*, Elsevier, July 1997.
6. M. O'Neill, M. Denos, "Automated System For Coarse-To-Fine Pyramidal Area Correlation Stereo Matching", *Image and Vision Computing, vol.14*, 1996, pp. 225-236.
7. O. Faugeras et. others, "Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms", *in Forstner and Ruweidel, eds.*, *Robust Computer Vision*, pp. 1-26, Wichmann, (1992).
8. H. Hseu, A. Bhalerao, R. Wilson, "Image Matching Based On The Co-occurrence Matrix", 1999.
9. R. Gonzalez, R. Woods,"Digital Image Processing", *Addison-Wesley*, 1993.
10. D. Ballard, C. Brown,"Computer Vision", *Prentice-Hall, Inc.*, 1982.
11. C. Sun, "Multi-Resolution Rectangular Subregioning Stereo Matching Using Fast Correlation and Dynamic Programming Techniques", August 1998.
12. R. Schalkoff,"Digital Image Processing and Computer Vision", *John Wiley & Sons, Inc.*, 1989.